Identity Verification Schemes for Public Transport Ticketing with NFC Phones

Sandeep Tamrakar
Aalto University &
Nokia Research Center
Espoo, Finland
sandeep.tamrakar@aalto.fi

Jan-Erik Ekberg
Nokia Research Center
Helsinki, Finland
jan-erik.ekberg@nokia.com

N. Asokan
Nokia Research Center
Helsinki, Finland
n.asokan@nokia.com

ABSTRACT

Public transport ticketing with mobile phones has in recent years become a possible reality as the standards for Near-Field Communications (NFC) are being taken up in mass transport ticketing, and the use of contactless smartcards for small value payments like ticketing is as well being deployed. We examine the feasibility of using mobile phone with a hardware Trusted Execution Environment for identity verification of transport ticketing with a perspective focusing on security and performance. We provide measurements based on an implementation in contemporary mobile phone hardware, and discuss our results by comparing with other proposed identity-verification ticketing solutions in light of the constraints set by usability and practical considerations as indicated by transport authorities.

Categories and Subject Descriptors

C.4 [**Computer-Communication Networks**]: Network Protocols — Authentication and security; D.2 [**Software Engineering**]: Design — Authentication and security

General Terms

Security, Design

Keywords

Public Transport Ticketing, Identity verification schemes, Authentication, NFC phone

1. INTRODUCTION

Public transport ticketing with contactless smartcards, based on ISO / IEC 14443 contactless card standard [12] as well as FeliCa[1] system, is widely adopted. With the expected widespread introduction of Near Field Communication (NFC) [13] capability on mobile phones, there is increasing interest in implementing public transport ticketing on mobile phones. A report by Juniper Research [24] predicts that by 2013 over 400 million mobile subscribers worldwide will use their mobile phone for ticketing. The primary reason for this is the possibility for significantly increasing the overall user experience of public transport ticketing because of the inherent communication and user input and output features on mobile phones. Such features unlock the ability to purchase and use tickets even while traveling abroad, reviewing past transactions, monitoring the remaining amount of time, value or other ticket properties, and the ability to easily pay for multiple travelers with a single interaction.

Public transport authorities have been particularly interested in *identity-verification ticketing schemes*, e.g. EMV [2] contactless card, in which a traveler can prove his identity (ID) to public transport check-points, located at the beginning, transfer points, and end of the journey. The transport authority can use such identity verification events to appropriately rate and charge the user [21, 17]. The basis of the identity-verification transaction depends on an *identity provider* who provisions an identity to the user's device, for example by having the user enroll a public key subject to some form of user authentication. Telecom operators, credit card companies, online service provider such as paypal could take the role of the identity provider. For simplicity, we also assume that the identity provider has a billing relationship with the user. In other words, the identity provider is both responsible for the security of the identity verification mechanism and carries the liability for journeys made by users whose identities have been verified.

The design of an identity-verification ticketing scheme has to simultaneously meet both functionality and security goals. The primary functionality goal is a definite time budget for an identity-verification transaction. In the case of public transport systems, we target an upper bound of 300 milliseconds (ms) as indicated by Smart Card Alliance [22]. The primary security goal, arising from the accountability requirement of the identity provider is to ensure that an attacker cannot successfully claim the identity of a legitimate user. In addition, as in any identity-verification system, it is desirable to minimize the ability of a third party to link the movements and transactions of a user.

The straight-forward identification solution is to have the identity provider issue long-lived identity certificates to user devices and to engage user devices in a challenge-response protocol with transport checkpoints where the response is a digital signature. The NFC reader we used (ACR 122U[3])

[1]Felicity Card - `www.sony.net/Products/felica`

[2]Europay, Mastercard and VISA `www.emvco.com`

[3]ACR 122U - `www.acs.com.hk`

STC'11, October 17, 2011, Chicago, Illinois, USA.

in our work incurred nearly one millisecond marginal cost for every additional byte transferred during the identity-verification session. Clearly, a straight-forward solution will fail to meet the target time budget.

While the available bandwidth for a particular technology will likely improve over time, the general problem of channels with tight bandwidth limitations will continue, for example with various sensor network radio technologies. Tight time-budgets for specific services that use identity-verification ticketing is a usability issue, and thus will also remain. This is the motivation for our work. We explore the design space for identity-verification ticketing solutions and study the effects of minimizing the data transfer on the security, privacy and other aspects of the system.

In this paper, we present an architecture and implementation for identity verification ticketing. We present two approaches to reduce the identity-verification transaction time. All our schemes use standard and widely deployed cryptographic mechanisms for authenticating, certifying and signing. All the cryptographic operations on the NFC-capable mobile phone are executed inside its secure, isolated Trusted Execution Environment (TEE). Our testing indicates that the communication speed rather than cryptographic computation will be the main bottleneck with the NFC interfaces in the currently available NFC hardware. Therefore, our attempts to reduce transaction times focuses on optimizing message sizes and minimizing number of messages.

We begin in Sections 2 and 3 by providing background knowledge and listing related work. System requirements are presented in Section 4, followed by an architecture overview in Section 5. Protocol variants are outlined in Section 6, followed by implementation notes in Section 7, and security and performance analysis in Section 8. In Section 9 we discuss about future improvements and finally conclude in Section 10.

2. BACKGROUND

In this section, we present various background technologies and concepts relevant for the ticketing implementation.

2.1 Trusted Hardware

For electronic ticketing, the default hardware element is the smart card, typically adhering to the ISO / IEC 7816 [15] interface primitives and to ISO / IEC 14443 [12] for the wireless interface. Smart card security as a rule follows the GlobalPlatform Card Specification standard [11], which defines the key management and provisioning protocols for compliant cards. Application / credential development for smart cards is predominantly done with the JavaCard programming language / toolchain. In some mobile phones (e.g. Nokia 6131, Nexus S) there is a built-in "embedded" smart card which is directly connected to the NFC radio chipset (e.g. NXP PN65 in Nexus S [19]). All of our ticketing algorithms can potentially be implemented using such hardware.

During the last decade, several types of TEEs have emerged based on the general-purpose secure hardware. These have been incorporated into mobile phones, and are nowadays widely deployed. A number of designs like Mobile Trusted Modules (MTM) [6], M-Shield [23] and ARM TrustZone [2] are available. These are either hardware agnostic (e.g. MTM), or augment the processing core for increased security. The latter can be typically combined with isolated RAM and ROM residing within the application-specific integrated circuit (ASIC) as well as with some amount of chip-specific "write-once —read many times" memory, typically implemented with E-fuses.

The TEE chosen for this work is the On-board Credentials (ObC) architecture [16]. In Nokia C7, ObC uses ARM TrustZone on a processor manufactured by Texas Instruments. ObC relies on the underlying hardware to isolate credentials from the operating system. Additionally, it provides a provisioning system and a byte-code interpreter with an extensive cryptographic Application Programming Interface (API) for the implementation of credential algorithms (ObC programs). The implementation also includes hardware-specific and proprietary mechanisms for generating a device key with ObC, certified by a Certification Authority (CA).

2.2 Transport tickets and security

As this paper describes a prototype system intended to be used for tests and trials in live ticketing environments, the realities of ticketed mass transport in general must be well understood. Mayes et. al. [20] provide a detail topical introduction to ticketing and fraud control. In terms of revenue loss the main system fraud are individuals that enter and exit the system without paying. Thus, it is not surprising that the biggest reduction in public transport fraud happened with the introduction of reliable gates and machine-readable tickets or physical tokens, since this eliminates traveling without a ticket or e.g. with a too cheap ticket.

Magnetic stripe technology is still widely used for ticketing although the security properties of that read-writable media is abysmal especially as the the transport gate readers, for technical and usability reasons, are not able to perform online-checks at the time of user entry or exit. Still, Mayes et. al. [20] point out that for one-time or limited period tickets a magnetic stripe or an unprotected memory card that "only presents a valid ID" can still possibly be considered sufficient, even in terms of security.

However, for more complex transport ticketing, where a transport identity is tracked through the system and fares are calculated based on use frequency and the exact journey that was undertaken, a more long-lasting relationship is constructed between a long-lived ID and its billing data. This also adds the customer (with the ID) as a potential victim of the fraud, since the use of a copied transport ID may accumulate additional charges for the original customer. To protect the users' interests as well as those of the transport authority, new ID-verification systems should provide some level of authentication to back up the identity claimed at the gate.

2.3 NFC

NFC is a wireless Radio Frequency Identification (RFID) technology standardized in ISO / IEC 18092 [13] and ISO / IEC 21481 [14]. An industry consortium, the NFC forum [4], provides compliance-testing and additional standards for NFC use. NFC devices support one or several of so called Type A, Type B and Felica coding, modulation and transmission schemes. Felica is used mostly in Japan and has a nominal transmission speed of 424 kilo bits per second (kbps) whereas types A and B are the norm in Europe and north America (106/212 kbps). The Nokia phones used in this prototype support all transmission options, but readers readily

[4] `www.nfc-forum.org`

available in Europe (also for transport) are limited to the lower transmission speeds. Section 7 will also show that the throughput on higher NFC protocol layers is in any case only a small fraction of the nominal speed provided by the radio hardware — a typical situation with wireless communication standards.

An NFC passive device is either a tag or a contactless card (ISO / IEC 14443), and gets its power from a reader. Active NFC devices (with their own battery) can operate in: 1) card reader / writer mode where they are communicating with a passive device, 2) in peer-to-peer mode when they are communicating with another active device, or 3) in card-emulation mode, where the active device "emulates" a passive element, e.g. a contactless card. Contactless cards provides the same ISO / IEC 7816-4 base command set independently of whether they are accessed over the wire (ISO / IEC 7816-1) or over NFC (ISO / IEC 14443).

We had the choice to implement the phone logic either using NFC card emulation or the peer-to-peer (P2P) mode. We chose P2P as readers are widely starting to support it, and we were then free to design our protocol and data units to interact also with TEEs that do not adhere to ISO / IEC 7186 specifications.

The lower layers of NFC include no communication security primitives. It is also well known that NFC technology is susceptible to e.g. both eavesdropping and man-in-the middle attacks [4], despite the fact that NFC is a short-range radio technology.

To date, some 20 phone models from different manufacturers support NFC[5]. The minimal operation available across all models is reader / writer mode. Only a handful of models embed a secure element or TEE that can be used for securing a ticketing transaction.

3. RELATED WORK

MiFare[6] is a contactless smartcard technology based on ISO / IEC 14443A developed by Philips / NXP semiconductors. MiFare cards are generally used for public transport ticketing, access control and event ticketing. Oyster card and ORCA card[7] are some of the MiFare based cards used in public transportation. Many attacks on MiFare Classic, a widely deployed member of MiFare family, has been published [5, 3, 8]. Further, the proprietary encryption primitive, CRYPTO1, has been reverse engineered and exploited to the point that the secret key can be extracted wirelessly in less than a second using ordinary hardware [9]. A study by Gans et. al [3] claims that the attacks are not applicable to the other members of the MiFare family.

FeliCa system developed by Sony Corporation is a contactless smart card for public transport, used primarily in Asia. In addition to transport ticketing, FeliCa based cards are used as electronic money for general purchases, loyalty cards, event tickets and even as credit cards.

EMV is a global standard that defines Integrated Circuit Card specifications for chip-based payment cards and also interoperability and compatibility of devices that accepts them. EMV specifications are developed and maintained by EMVCo — a joint effort of Europay, MasterCard and Visa.

EMV has published standards for contactless payment using ISO/IEC 14443 cards [7]. EMV based cards issued by financial institutions such as U.S. Bank, Barclaycard and Clear2Pay[8] are not only used for basic payment but also combine functionality for public transportation, loyalty cards and event ticketing.

The EMV specifies three methods of offline data authentication techniques: Static Data Authentication (SDA), Dynamic Data Authentication (DDA) and Combined DDA and application cryptogram generation (CDA).

In SDA, the card provides the reader with two long-term, size-optimized public key certificates — an issuer attribute certificate and the issuer's public key certificates signed by a certificate authority. The readers only verifies the certificates and do not send any nonce during authentication. Thus, this method is vulnerable to replay attacks and card copying.

The DDA method authenticates cards using a challenge-response technique. This method assumes that each card holds a unique pair of keys signed / certified by the issuer. A recent study by Anderson et. al. [1] mentions that a single DDA signature operation on a Visa card takes around 426 ms and up to 714 ms on MasterCard.

Since 2001, Short Message Service (SMS) based public transport ticketing has been launched by Helsinki city public transport. Later, similar services have been offered in cities around the world (e.g. Prague, Rome). The ticket is purchased by sending an SMS request to a ticketing server which then issues ticket in an acknowledgment SMS. Such systems are vulnerable to locally generated SMSs with predefined sender, recipient, body and status flags [18].

NFC ticketing [10]— a project by RFID Lab of the University of Rome "Sapienza" provides a public transport ticketing on an NFC enable mobile phone. The ticket information is managed and stored by a JavaCard application running on the secure element of the phone. A midlet application provides a user interface to purchase the ticket and to display ticket information. Ticket acquiring is performed over SMS. Their focus is not on security but on usability issues such as purchasing ticket and using "pay as you go" model.

4. REQUIREMENTS

A transport ticketing system consists of many functional elements, devices / cards and (business) stakeholders. In this paper, we only examine a subset of a system where the end user protocol is ultimately based on terminal identification. On a high level, such a system includes an *accounting / certifying* authority (CA) which provisions a customer / transport identity ID_D to the mobile phone (user device D). CA is the one that collects money from the user and forwards that value to the *transport* authority based on the transport service consumed by corresponding user. We represent transport authority including its backend infrastructure as TA. At each transport check-point, located at entry, transfer and exit gates, the D provides its ID_D to a ticket reader (R) operated by the transport authority. Based on the identity received and service used, TA runs a fare engine that maps, for the customer, his transport service consumption to a price, which is then invoiced through CA. The accounting authority CA can operate prepaid accounts or further bills its customers for public transport usage.

[5] A list of NFC-enabled phones can be found at `http://en.wikipedia.org/wiki/Near_field_communication`

[6] `www.mifare.net`

[7] `www.oystercard.com`, `www.orcacard.com`

[8] `www.usbank.com`, `www.barclaycard.co.uk`, `www.clear2pay.com`

In this section, we outline the essential requirements for the identity-verification transactions within the transport system. We present three types of requirements: *functional* requirements which determine the operational constraints as well as the targeted level of non repudiation, a *privacy* requirement that addresses device tracking in the system, and finally the *accounting* requirements give a fundamental on which liability in the system can be determined.

4.1 Functional Requirements

At a transport check-point, a ticket reader R must be able to read the transport identity ID_D of each device D that "taps" it. The reader R must be able to validate the authenticity and eligibility of the ID_D before allowing access to its service.

$D \longleftrightarrow R$ represents a successful user authentication session between a device D and a reader R. A limited blacklist should be provided by CA to TA, and processed during each $D \longleftrightarrow R$ session at R in real-time. Further, depending on protocol, CA may need to provide an interface by which TA can validate the eligibility of the ID_D. Such validation may be required before processing an identity that is previously not known to the transport system.

The total time taken by any successful $D \longleftrightarrow R$ session must not exceed 300 ms. And the information collected by the reader R during such authentication session must be stored as an evidence and later made available to the transport back-end system TA for rating service fares.

Also, the transport authorities have indicated a reluctance to store and manage any secrets in their ticket readers. This is foremost a practical concern, involving mitigating the risk of ticket readers being stolen for secret recovery as well as realizing that the management of secrets also implies running cryptographic protocols inside the network of the transport authority.

4.2 Privacy Requirement

An eavesdropper must not learn an ID_D of a user from any $D \longleftrightarrow R$ session. The privacy statement is intended to protect user tracking. One way to achieve privacy is to have variable transport identities for each user. However, variable IDs must not negatively affect the transport service fare rating performed by the transport authority.

4.3 Accounting Requirements

The transport authority backend server TA needs to have access to identity verification evidence for associating invoicing information to that evidence before sending it to CA. Later, CA collects service charges on behalf of the transport authority based on the evidence and in accordance to the invoicing information. The CA is at least partly liable for the fraud where charges are associated to a wrong customer or ID_D.

On the other hand, device D must also store $D \longleftrightarrow R$ session evidence in case of possible ticket inspection. Collecting such evidence can also be used from device application for viewing traveling history.

These requirements further imply that:

- The ticket reader R must ensure the data elements exchanged in a $D \longleftrightarrow R$ session is according to the instruction provided by CA. Only evidence appropriately validated during the real time data transaction will be accepted later by CA.

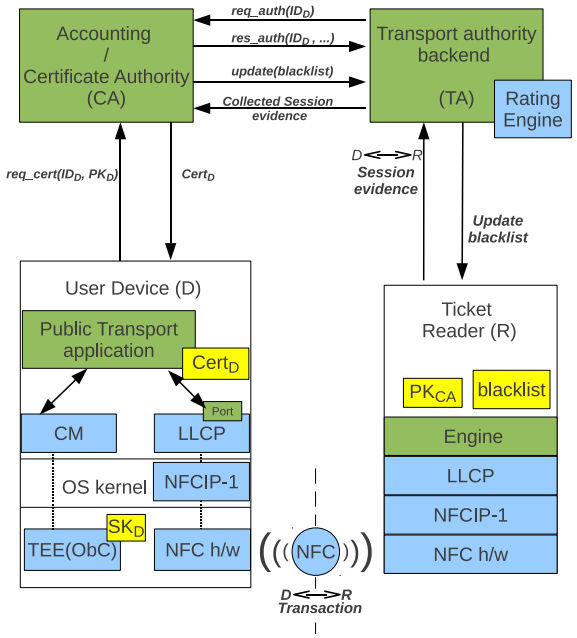

Figure 1: Architectural overview

- CA must provide an interface by which TA submits evidence information and invoices.

- CA is responsible for making available, for the transport authority, a blacklist based on (temporarily) "disconnected" ID_D, e.g. due to pre-paid value running out, abuse of the identity, lost credit rating or on customer request.

5. ARCHITECTURE

The identity-verification ticketing architecture is shown in Figure 1. The mobile phone and its customer has an a-priori relationship with a service ecosystem. The accounting authority CA of that ecosystem is able to uniquely identify the device D and its secure element (TEE), and is also responsible for billing / charging customer.

For the purpose of transport ticketing, the TEE of the device generates a new RSA keypair, the private part of which will never leave the TEE unencrypted. The public component is sent to the accounting / certificate authority in the transaction $D \longleftrightarrow CA$, whereby it is certified for use in the ticketing context. Our improved protocol variants also adds further information into the $D \longleftrightarrow CA$ and / or some of its secret data. Depending on the variant, $D \longleftrightarrow CA$ is thus either a key certification operation, a secret provisioning operation, or a combination of both.

The returned certificate $Cert_D$ will always contain a unique "transport ID" ID_D to be used in the transport context with one or several transport authorities. The ticket readers at transport system check-points are provisioned with necessary algorithms for the identity verification as well as the public component PK_{CA} of the accounting authority's secret key which is required to validate the above certificate.

When the mobile phone touches a ticket reader at a check-point, an identity-verification transaction $D \longleftrightarrow R$ is carried out. Section 6 outlines three variants of this protocol. As part of the transaction, the check-point not only validates the device identity certificate and possibly the signed challenge response, but also validates the received transport ID against a blacklist.

The transaction evidence, i.e. the reader's challenge, the received certificates, signed responses, transaction time and location, is collected along with other system-internal data. This evidence will be further fed to a rating engine and finally submitted to the accounting authority for invoicing at the $TA \longleftrightarrow CA$ interface.

The system also needs to support a mechanism for blacklisting identities during the validity of their respective transport certificates. Of course, no further transport certificates will be issued by the CA to the device over the $D \longleftrightarrow CA$ interface until the cause for the blacklisting is resolved. The exact usage context of the blacklisting is transport operator dependent, but the basic intuition is that the fare collection liability of the CA ceases for a device D when it is entered on the blacklist.

In the following subsections we define the key entities involved in this system.

5.1 Accounting / Certificate Authority (CA)

CA represents the accounting authority that issues identity certificates to user devices and maintains billing information. In addition to this, CA is also the entity responsible for clearing the transaction and maintaining a blacklist for devices or users who do not fulfill their monetary obligations or have been subject to identity-theft.

In order to generate device certificates, CA requires a signing facility with a service private key (SK_{CA}) and CA must provide a corresponding public key (PK_{CA}) to its relying parties for verification.

5.2 User Device (D)

D represents any NFC integrated mobile phone that will be cryptographically associated with a given account holder by CA. D contains a secure, isolated trusted execution environment for processing confidential data. In identity-verification ticketing schemes, this typically amounts to a hardware-rooted functionality for producing signatures with a key protected by the very same TEE, e.g. a device private key (SK_D) corresponding to a public key (PK_D).

Each device representing a user will be associated with a customer / transport identity (ID_D), assigned by CA. However, its format is either transport authority dependent, or accepted by the transport authority for use in its processing.

The user identity certificates ($Cert_D$) issued by CA on a regular basis includes information such as PK_D, ID_D and the expiry time (T_{exp}).

5.3 Ticket Readers (R)

R represents an NFC ticket reader attached to the transport system gates. To validate $Cert_D$ of the user device D, each reader needs to have access to PK_{CA} — the public key of the accounting / certificate authority CA. Each reader R possesses a unique, possibly cryptographic, identity ID_R. ID_R may also be share by all checkpoints located at the same station.

Readers also contain an interface to receive continuously updated blacklist information for user identity verification. The readers also gather all evidence related to each $D \longleftrightarrow R$ transaction. This information eventually will be needed for fare calculation and auditing by CA.

5.4 Transport Authority Backend Server (TA)

TA is the representation of the backend processing system of the transport authority. This system is responsible for providing the transport services to the users. In the context of the ticketing scheme, it is responsible for operating the ticket readers and gates as well as collecting all evidence from all $D \longleftrightarrow R$ transactions in the system, performing fare calculations for the users, and submitting the evidence and the invoicing information to CA. TA also queries the blacklist from CA and distributes it to its ticket readers.

Any data transaction between backend server TA and CA shall not be synchronous to the data transaction between D and R. In some business scenarios near-real-time interaction might be needed though, e.g. a special blacklist query $TA \longleftrightarrow CA$ may follow soon after a $D \longleftrightarrow R$, e.g. the first time a user with a device D uses a given transport system.

6. PROTOCOLS

In this section we present three different protocols for authenticating the user device D at a ticket reader R. We assume that cryptographic operations on the mobile device are carried out inside a TEE or secure element. For the ticket reader, only functional integrity is assumed for most protocol features (privacy-protection needs some basic level of confidentiality).

The baseline variant called the "Standard protocol", is the straight-forward certificate-based authentication scheme. Our measurements (Section 7) indicate that its implementation is infeasible in contemporary devices within the given functional constraints. Hardware acceleration support for more space-efficient algorithms, like elliptic curve cryptography, in the readers and improvements in NFC transmission speeds could solve this problem, but these are absent in the NFC readers commonly available today.

The two variants described in Sections 6.2 and 6.3, describe ways of improving the baseline variant that can meet the functional and security requirements using cryptographic primitives widely available in readers.

6.1 Standard protocol

In the standard protocol, a device D receives a long-lived identity certificate $Cert_D$ from CA. The validity of such a certificate is in the range of months, i.e. the user's device D will contact CA over GPRS/3G or WLAN a number of times per year to receive a new identity certificate. This identity certificate holds information such as public key of the device PK_D, the customer identity ID_D, and expiry time T_{exp}.

From measurements, it is evident that a standards-compliant X.509 certificate cannot be used because of its size. In this paper, we use an optimized certificate for an RSA public key with elements T_{exp}(4 bytes), ID_D (6 bytes), $PK_D(modB$ bytes) and two bytes of additional ticket and protocol-related information in addition to the signature, i.e. $12 + 2modB$ bytes, where $modB$ is the size of the RSA modulus in bytes. This in contrast to around 3-400 bytes of overhead (in addition to $2modB$) that a full X.509 certificate entails.

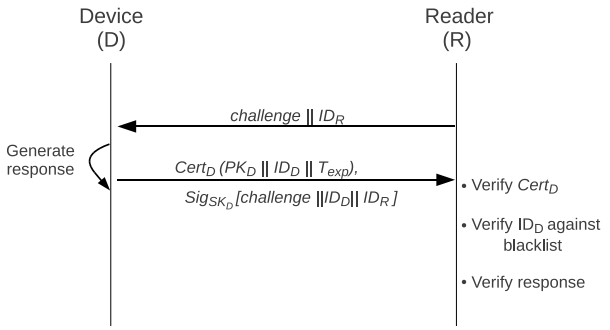

Figure 2: Standard Protocol: Messages in $D \longleftrightarrow R$ session

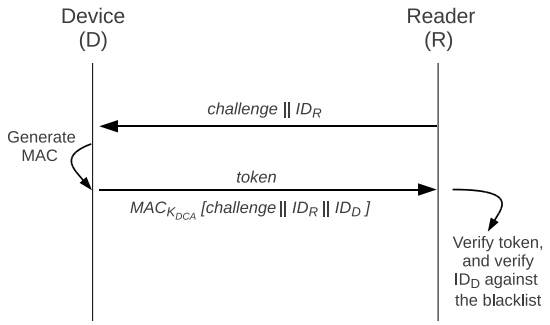

Figure 3: Variant 1: Messages in $D \longleftrightarrow R$ session

For the ticket reader R, we assume a unique identity ID_R and the knowledge of the public key PK_{CA} of the CA. Once a device D comes within the range of R, they engage in a quite traditional challenge-response interaction, as shown in Figure: 2:

When a user wants to enter a station gate, he or she taps his or her device D to a reader R attached to the station gate. R sends a random challenge *challenge* along with its identity ID_R. D, on receiving the *challenge* computes a *signature* over the challenge and the reader's identity, i.e., $Sig_{SK_D}[challenge||ID_D||ID_R]$. It sends back this signature, along with its certificate $Cert_D$ to R. R validates $Cert_D$ using PK_{CA} and ID_D against its blacklist. If the ID_D is blacklisted, it simply denies service access to D and terminates the session. Otherwise, if *signature* is correct, it allows access to D. The full transaction information is combined with potential auxiliary context information as *evidence* and eventually sent to the CA.

The standard variant can be a good method for realizing an identity verification ticketing scheme. More or less all TEEs includes RSA key generation and signature capability as default functions, making the protocol viable on mobile hardware ranging from contactless smart cards powered from the reader to more stand-alone devices with TPMs, MTMs and TEEs. The primary drawback is, as we will see in Section 7, the difficulty of meeting the time budget.

6.2 Variant 1 — Short-lived certificates with MAC

In this protocol variant, a device D receives an ephemeral attribute certificate as an access *token* from CA. The token remains valid for a few hours. In this model, the user device D is expected to contact CA relatively often to receive new certificates. Unlike long-lived certificates, the *token* includes a complete validity period — T_{begin} (4 bytes) that indicates the time from which the *token* can be used and T_{exp} (4 bytes) the expiry time of the token. It also consist of transport identity ID_D (6 bytes) and two bytes of additional ticket and protocol-related information in addition to the signature but does not contain device public key information. This significantly reduces the size of the token to $(16 + 1 mod B bytes)$, compared to a full identity certificate.

The reader R still stores PK_{CA} for verifying the tokens, but the identity verification omits validating the device signature — they are still received and stored though, and the signature is still conveyed from TA to CA for later auditing.

Since this signature need not be verifiable by R, it can be made even smaller (i.e. 20 bytes) by constructing it as a message authentication code (MAC) using a symmetric key K_{DCA} shared between D and CA.

An overview of this protocol variant is depicted in Figure: 3. The message transaction sequence is outlined in the following:

1. When a user wants to enter through a gate at a station, he or she taps R with D.
2. R establishes NFC communication with D and initiates the session by sending a random challenge, along with its identity ID_R and other context related data.
3. On receiving the *challenge* the device D immediately returns the *token* to the reader R.
4. In parallel, D now computes a MAC ($MAC_{K_{DCA}}$), inside its TEE, on the received challenge using a shared key K_{DCA} provisioned to D by the CA. Once computed, $MAC_{K_{DCA}}$ is also sent to the reader R.
5. Immediately after receiving the token, reader R verifies it using PK_{CA} and also verifies the ID_D of the device against its blacklist. The validity of the certificate is of special interest in this protocol variant — if the certificate is given too early or too later or if the ID_D is blacklisted, R denies the user entry and terminates the session.
6. After verifying ID_D, R allows access to D. It also collects all transaction evidence, combines it with auxiliary context information. Further protocol steps in no way differs from the standard variant.

In this protocol, the device D only shares the secret key to the MAC with CA. R is not expected to verify the MAC, it only stores it with other evidence for further transmission to CA, via TA for verification. After computing the signature over the received challenge, the device D may also directly send it to CA for later auditing and verification. This variant only involves verification of the token and ID_D and omits the transmission of PK_D during each $D \longleftrightarrow R$ session.

The advantage of this method is that the response sent during identity verification is even shorter. Also, the cryptographic schemes used are all standard. The disadvantage

is that, since the *token* is the only cryptographic entity verified by R, the system is susceptible to replay attack i.e. an attacker can replay *token* to gain admission to the system. This can be reduced in a couple of ways:

a. Include data in *token* limiting the scope of its applicability (e.g. identities of the checkpoints — in the public transportation case, the identities of the particular stations that the user typically uses.)

b. Limit the lifetime of the *token*. However, limiting the lifetime increases at least the number of tokens needed to be produced for a given device D. It is conceivable that D can get a batch of attribute certificates in one connection to CA, say enough to last a week or two.

A rough level of time synchronization is implicitly implied by protocol variant 1, although this is unlikely to be an issue in a deployment - CA and Rs can be assumed to have accurate time information, and in fact, so do most devices D. Even if this is not the case, the device owner can adjust the time of D to be correct in case of a failed authentication.

Depending on the ticketing policy and associated rating logic, the frequent change of customer tokens can also be used for privacy. In case the transport authority does not need a unique token for rating (or if it is acceptable that the rating utilizes an identity resolving service provided by CA), then the device identity ID_D can be varied even with every token, making the tokens of one customer indistinguishable from those belonging to another from the perspective of an eavesdropper at the D to R interface.

6.3 Variant 2

This protocol variant uses the same solution as described in Section: 6.2, but incorporates the end of a hash chain[9] in the attribute certificate to primarily reduce the frequency of fetching tokens, but if the transport authority can deploy some form of lightweight near-real-time distribution of recently used hash chain elements among all its readers R, it can also serve as an effective replay protection mechanism.

Similar to *token* in Section: 6.2, the device D receives a short-lived certificate $Cert_D$ from CA. The only difference is that, this certificate also contains the last element c_o (16 bytes) of the hash chain to be revealed by CA in reverse order. Thus, the size of the certificate is increased by additional 16 bytes: $(16 + 16 + 1 mod B) bytes$. Similarly, the reader R still stores the public key PK_{CA} of the certificate authority for verifying the certificate.

An overview of this protocol variant is depicted in Figure: 4. The message transaction sequence details are outlined below:

1. When a user wants to enter through a gate at a station, he or she taps R with D.

[9] Here, the reverse hash chain c is used as access token. A user device D generates a reverse hash chain using a repeated n times hash on some random number x. Then, the first element c_o of the reverse hash chain is the n^{th} hash value, second element c_1 is the $(n-1)^{th}$ hash value and so on ending the chain with the random number x. As each element of the chain is revealed, it can be verified as belonging to the same chain since the hash value of the element result in the previous element of the chain. We represent elements of the chain with c_i, where i represents the ordinal of the element. In order words, i represents the distance between the first element c_o to the i^{th} element c_i of the chain.

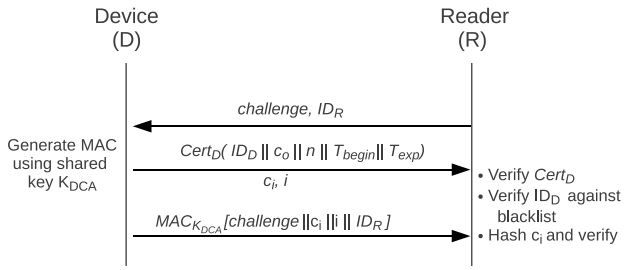

Figure 4: Variant 2: Messages in $D \longleftrightarrow R$ session

2. R establishes NFC communication with D and initiates the session by sending a random challenge along with its identity ID_R and other context related data to D.

3. On receiving the challenge, D sends back its identity certificate $Cert_D$ along with the current element in the hash chain c_i and the value of i, i.e the distance between the first element c_o to the c_i.

4. In the mean time, D generates $MAC_{K_{DCA}}$ on the identity of the reader ID_R, the *challenge*, the element of the chain that was sent to the reader c_i, and the value of i. Once computed, this is also sent to the reader R. However, $MAC_{K_{DCA}}$ is not expected to be verified by the reader R but to be forwarded to CA for verification and auditing.

5. On receiving the identity certificate and the current hash chain element c_i, R verifies the identity certificate, as well as c_i by hashing it i times and comparing it with c_o. The value of i is validated according to a policy defined later. R also verifies ID_D against the blacklist.

6. Once verified, R allows access to D, collects all the information, combines it with an auxiliary context information as evidence for the rating procedure and protection against repudiation at a later time. This information may be sent in a batch or in real-time towards the CA.

There are several ways in which the hash chain (the c_is) can be used for mitigating replay attacks:

a. Limit the duration of validity of c_i; This requires loosely synchronized clocks between D and R. The validity duration can be made rather small (e.g. 5 minutes). The smaller the validity interval is made, the smaller the time window for a successful replay attack gets. The advantage of the window size is debatable, as an eavesdropped value certainly can be re-used and even transferred between stations for re-use within a matter of seconds at least.

b. Have R report to a common channel the latest hash value to a centralized backend after every accepted identity verification. Before accepting an identity verification for a given $Cert_D$, R should check that the claimed i value has not already been used. Even for

large transport systems like the London Underground, where in peak days around 3 million customers uses the transportation service, the new hash values (16 bytes assumed) that need to be spread out across the transport system amounts to a meager stream of 10 kbps. As this data is used only for replay protection and not necessarily critical if overflowing or partially lost, incorporating such a system with readers R is certainly possible. The data set (even for a whole day) in such a system is of course at worst a manageable 50 MB.

c. Combining the two earlier mechanisms, short time intervals limits the size of the dynamic look-up table needed in readers R and it also lessens the risk of accidental value collisions – which is minimal to begin with. For 5 minute intervals, in a system like the London Underground, the cache of recently used hash values can be limited to around 300-600 KB, as they anyway expire after 5 minutes.

The main advantage of variant 2 is that the response sent during identity verification is as short as in variant 1, but the threats of replay attack can be limited without increasing the frequency of identity provisioning transactions.

In case if the hash chain values can be distributed in the transport system network, then a similar threat mitigation policy can be deployed for all variants by keeping track of IDs which are "in the system", and which are not. As gated systems can separate between entries and exit, the use of eavesdropped data becomes significantly more difficult if exit alerts are sounded and entry is denied for tap actions where the ID_D is already assumed to be in the system at entry, alternative not exit at present.

6.4 Backend data transactions

CA is responsible for maintaining blacklist information for all its customers based on payment history and credit / debit balance. In the long term, customers whose credit rating is too low, will not receive transport certificates from CA. However, since the certificates are long-lived, their revocation in the short term is handled by blacklists.

Figure : 5 shows the data interfaces between back-end servers TA and CA. The assumption is that devices with a valid transport certificate are allowed at least the first trip, while the back-end interfaces are invoked.

- An authorization interface provided by CA to TA gives the real-time state of the potential blacklisting for a given ID_D. Thus, when a new customer enters the system, the customer will be let in based on the existence of a fresh certificate for his ID_D. However, the transport authority will immediately or soon check the state of that customer at the authorization interface. Only a successful authorization moves the liability for the travel of that ID_D to be handled by CA.

- For ID_Ds used in a transport system, periodic blacklists will be provided to improve the efficiency of CA to TA communication.

- TA will process and rate the travel related to an ID_D independently of CA. TA is also required to temporarily store all evidence from $D \longleftrightarrow R$ sessions that relate to the traveling done using the device D as a "payment instrument". Periodically, e.g. once a day, TA will present the accumulated information, i.e. the rating and the evidence, to CA for accounting and further storage.

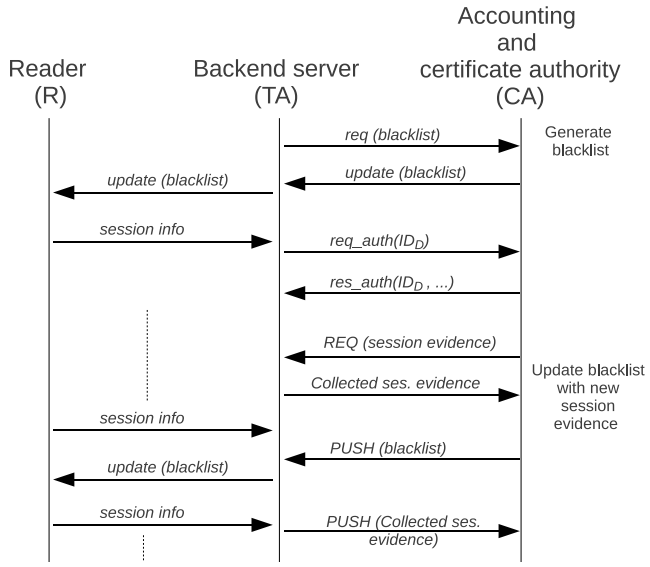

Figure 5: Messages in a backend data transaction session.

7. IMPLEMENTATION

For our implementation we used readily available hardwares and tools. We choose Nokia C7, running Symbian ∧ 3 operating system, as our user device D. It includes necessary hardware and software support for both the TEE and NFC capabilities. We implemented the ticketing application on the phone using Qt APIs[10] that invokes Qt mobility APIs for NFC communication and calls "ObC plugins" for all cryptographic operations like key generation, provisioning, signing, and hashing etc. For the phone to communicate with the ticket reader, we choose Logical Link Control Protocol (LLCP) connectionless mode of NFC that enables two NFC devices to exchange data. LLCP is implemented on top of NFCIP-1 and these constitute the peer-to-peer protocol stack in NFC.

The ticketing application provides all the necessary interfaces for user interaction such as checking travel history and monitoring remaining time or balance. The application also communicates with the accounting entity over the Internet not only for obtaining and updating transport identities and certificates but also for synchronizing data such as travel history, account balance, rating information on current travel etc. For topping up an account balance user can purchase new credits over a secure Internet connection using a web browser.

For the transport authority functionality we used a commercially available NFC reader ACR 122U. Later, we also tested our system using another NFC reader Ask LoGO[11]. The reader is connected to a Linux PC running Ubuntu

[10]Qt- `http://doc.qt.nokia.com`

[11]Ask LoGO is manufactured by Ask `www.ask-rfid.com`

Data size	ACR 122U	Ask LoGO	Nokia C7
48 bytes	16 kbps	24 kbps	8 kbps
64 bytes	16 kbps	24 kbps	12 kbps
128 bytes	18 kbps	30 kbps	12 kbps

Table 1: Comparison of LLCP throughput between a Nokia C7 phone and different NFC readers (baudrate set at 106 kbps) with different data sizes

Maverick that handles our protocols. We used an open-source project *libnfc*[12] for accessing NFC reader in our PC.

The libnfc provides peer-to-peer communication functionality between two readers up to the NFCIP-1 protocol level but does not provide any LLCP functionality. Thus, we implemented the LLCP communication protocol (particularly connectionless mode) based on LLCP specification published by NFC forum.

We also implemented a *reader application* on the PC that serves both ticket reader functionality as well as transport authority functionality. The application interacts with the user device and verifies data received from the user device like device identity certificate, signature over random challenge etc. After successful verification the application signals a relay connected to the PC that controls a demo gate.

The application then collects all the essential information such as transport identity ID_D, device identity certificate $Cert_D$, time, challenge sent and signed response as an evidence. At the current stage of our implementation, we do not yet have service rating strategies thus, we simply decrement the account balance by value 1 for each successful user session. The collected information is then sent to the accounting entity over a secure Internet connection.

Measurements

Before implementing our protocol we measured the throughput of LLCP connectionless communication between the phone and the readers. Additionally we also measured the LLCP throughput between two Nokia C7 phones. All the mentioned readers uses different chipsets: the ACR 122U uses NXP PN532, Ask LoGo uses NXP PN533 and the Nokia C7 uses NXP PN544. (The maximum payload allowed by PN532 chipset is 252 bytes of which some bytes are used by LLCP header). Thus, we sent three fixed size data blocks of 48 bytes, 64 bytes and 128 bytes from our readers to the phone and back. We measured the throughput based on Round Trip Time. For the measurements, we set the NFC modem of the readers to passive mode with a baudrate of 106 kbps.

Table 1 shows the average throughput calculated from 20 individual measurements. From our measurements, we found the highest achievable data rate is around 30 kbps. However, in order to implement our protocol in a real devices our application should account for slower data rates to be fail safe even in the worst case performance. Therefore we designed our system assuming average data rate to be 8 kbps.

Figure 6 shows the timing diagram of different events that occur during a successful $D \longleftrightarrow R$ session in the standard protocol described in Section 6.1. The figure is plotted by averaging measurements obtained from 20 different $D \longleftrightarrow R$ sessions between Nokia C7 and ACR 122U. All the cryptographic operations are based on 1024 bit RSA keys. Similarly Figure 7 shows the timing diagram of different events that occur during a successful $D \longleftrightarrow R$ session in the protocol variant 2 described in Section 6.3.

[12] `www.libnfc.org`

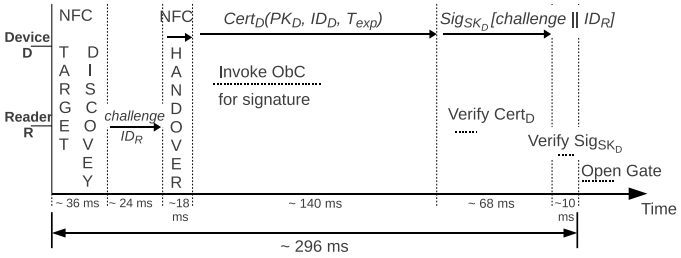

Figure 6: Total time taken for standard protocol with 1024-bit RSA key to complete a $D \longleftrightarrow R$

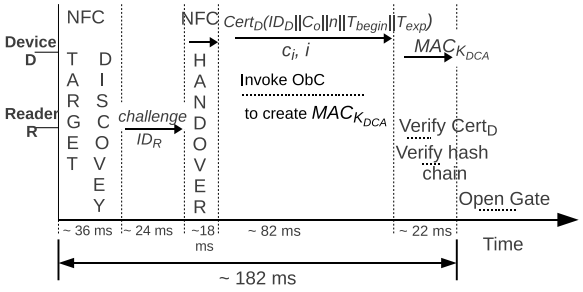

Figure 7: Estimated time required for Variant 2 with 1024-bit CA key to complete a $D \longleftrightarrow R$

The figures shows that on average it requires 36 ms since the "tap" to discover the target before the reader can actually start data transmission. The reader initiates the session by sending 50 bytes of challenge including the identity of the check-point reader. In return, the phone always sends the NFC connection handover message. This message is sent by the phone's NFC implementation by default whenever an NFC connection starts, there by introducing an 18 ms delay before identity certificate $Cert_D$ is transmitted back to the reader. The ObC signature invocation which takes around 64 ms (when measured from the Qt application) is executed in parallel to the $Cert_D$ transmission.

On receiving the $Cert_D$, the reader application validates the certificate immediately or in parallel while receiving the signed response. In standard protocol, the signed response is validated immediately after receiving whereas in variant 2 the signed response is forwarded to CA for validation.

Figure 6 and 7 indicates that the standard protocol for identity-verification barely satisfies the 300 ms threshold but the protocol variant 2 takes significantly less time to complete $D \longleftrightarrow R$ session. These measurements are for a 1024-bit RSA signature key and a certificate signed with a

RSA Key size	Standard Protocol	Variant 1	Variant 2
1024 bits	296 ms	164 ms	182 ms
1152 bits	314 ms	172 ms	190 ms
2048 bits	482 ms	228 ms	246 ms

Table 2: Average time taken by protocol variants to successfully complete a $D \longleftrightarrow R$ session.

1024-bit long-term CA key. 1024-bit keys are deprecated by EMV since 2009, and 1152-bit keys are still acceptable up to 2011. This difference account for some 20 ms extra delay in transaction, causing the protocol to miss its time constraint. Furthermore, should we deploy a full 2048-bit RSA CA key and 2048-bit RSA device key then we push the average transaction time into 480 ms range.

Table 2 shows the overall estimated time required by all three protocols using different RSA key length to complete a successful $D \longleftrightarrow R$ session. The time values displayed in the table do not account for the time required to open a gate. The values in the table clearly shows that the standard protocol would fail to meet the 300 ms time threshold when key size is increased. However, variant 1 and 2 would complete a $D \longleftrightarrow R$ session within 300 ms even with an RSA key size of 2048 bits.

8. ANALYSIS

In this section, we provide an informal analysis of the security properties achieved by our architecture.

The **isolation** of credential secrets, i.e. the RSA private key and its use, as well as the hash chain calculations, are guaranteed by the TEE and the ObC, as is outlined in [16]. The integrity of evidence data stored on the phone for user logging is protected by platform security in Symbian.

The **integrity of the protocol between D and R** stems from the fact that the device signature is calculated in the ObC. Further in all cases, data is bound to either the externally signed certificate or the locally signed signature, including the challenge and identity from R. In variants 1 and 2, the integrity of the device signature cannot be validated by R, which adds an attack where the *token* is copied between devices. This replay attack is made more difficult by shortening the lifespan of the token (variant 1) or by adding the hash chain as a way to further break down the reveal data in smaller quanta verifiable by R or the system behind it.

Man-in-the middle, or more specifically relay attacks, are possible to mount on all protocol variants, since there is no user interaction in the protocol - in many cases the ticketing application may even be always-on to improve usability. However, the reader equipment for transport ticketing exist only in well known and well guarded environments, and it is conceivable that the effort of mounting a relay attack for ticketing clearly out-weights possible monetary gain achieved by it. Contrary to contactless smart cards, the mobile phone has the option to activate and deactivate the ticketing / NFC interface based on context, e.g. according to location or the device lock being on or off, to limit exposure to the relay threat. Also, some of the replay protection mechanisms listed in Section 6, e.g. using the knowledge who is in, and who is out of the transport system, also limits the efficiency of the relay attack. Even a recognizable device "beep" when the identity-verification takes place may alert the user to him being subject to an attack.

Non-repudiation of the evidence for most parts also follows from the protocol set-up, and the evidence that the device D unconditionally signs. However, there is by default e.g. no audited log of the interaction time of the $D \longleftrightarrow R$ transaction, and considering that the device signature in variants 1 and 2 is symmetric-key based, non-repudiation relies on partial trusted relationship between D and CA, and to some degree also between CA and TA.

For the communication between D and CA, confidentiality and integrity of the data exchange is guaranteed by the ObC provisioning protocol [16]. D has a device certificate that can certify a provisioning key, as well as any other key (e.g. the ticketing challenge signing key) in the ObC environment of D. The provisioning protocol [16] also sets up a security domain combining provisioned algorithms and keys. Based on this, ObC provides domain isolation for provisioned secrets and programs endorsed to the ObC application domain, e.g. for the hash chain calculation, where the top of the chain is kept a secret.

Privacy, or more specifically, protection against identity tracking by eavesdropping, is a service that none of the presented variants solve in an acceptable manner. The baseline requirements that the transport authority must be able to do the rating (based on some consistent ID_D) and the requirement not to have secrets in the readers R, in combination with the time budget seems to make the problem intractable. One possible solution is to use very short-lived tokens or certificates with variable ID_Ds, and to add an additional transactional step between the reader activity and the fare engine where the actual transport ID is resolved from the temporary one. This complicates the overall protocol set, and we chose not to emphasize this option.

This work however does not address the user privacy against the backend systems; i.e. CA and TA can build a route information profile of a user based on the user's traveling behaviour. While such profiling is not desirable from a privacy perspective, it can be used to suggest alternative routes to the user during traffic congestion. Further, such information can also be used to implement flexible pricing policies such as congestion pricing, transfer discounts etc [21].

Time budget: As the primary reference to the identity-validation is an EMV contactless card, let's consider its performance. The DDA protocol is the closest match in our setting. Two long-term size-optimized certificates are first retrieved by the reader for validation. This alone breaks the 300 ms barrier with the readers we have experimented with, but if we assume that the card can operate at close to ideal 106 kbps speeds, the latency is around 50-100 ms. Then based on the measurements in the work [1] the signature takes more than 400 ms. Adding these measurements, we are at best looking at tap times of 500 ms, most likely significantly more. As a consequence, contemporary phones are clearly competitive as identity tokens for transport even when compared to the state-of-the art in card technology.

In Table 2, we estimated that deploying a full 2048-bit RSA key on a reasonable variant in terms of security would push the average transaction time into around 480 ms range. These time differences seem trivial at first sight, but with this time-extended interaction the required user activity on system entry and exit changes from a "tap" to a "tap-and-

hold", interrupting the flow of customers, especially as the transaction success rate also now plummets, caused by terminal movement during the transaction. In this light, our **preferred architecture variant** is variant 2, with hash chain elements used only for limiting the validity to short time intervals. This implies clock synchronization, and we add a reference clock in the server challenge for this. No additional signaling is needed in the backend network. This approach does not solve the threat of **relay attacks**, but mitigates the usefulness of ticket **replay**.

Since the system logs the complete transaction, the accounting authority or the fare engine can add auditing logic to estimate the amount of replay / relay attacks occurring in the system. As the phone security logic and the user application can be upgraded, as can reader software, a solution to the replay problem can be added to the system later, if needed.

9. FURTHER IMPROVEMENTS

The main limiting factor of protocol improvement is the communication bandwidth. If the speed of communication can be raised significantly, the next features to be implemented are: 1) better privacy protection based on reader public keys and 2) adding an authenticated and possibly secret channel between device D and CA to transfer also auxiliary information like a transaction receipt to the customer. Also, standard certificates for the CA could be deployed. Currently, none of these features are deployable.

10. CONCLUSIONS

In this paper, we presented an implementation of the mobile-phone relevant parts of an identity-based mass transport ticketing architecture. The end result meets the main functional requirements within the time budget of 300 ms. The architecture shows the advantage of using a mobile phone as a ticketing device both in the user interface integration, and also in presence of the communication channel between the phone and the accounting server, neither of which are present in contactless smart cards. References also indicate that contactless smart cards are not yet performing within the requirements for efficient public transport ticketing, neither in terms of performance, and plausibly not even in terms of security. Also for mobile-based ticketing our measurements indicate that performance bottlenecks in the communication speed for now clearly limits us in the aspiration to properly address privacy and replay attacks. Nevertheless, mobile phones have now reached a maturity level where they can be used for transport ticketing, and we believe that the industry has the potential to move from trials to actual deployment in the next few years.

11. ACKNOWLEDGMENTS

We thank thank Jukka Virtanen and Kari Kostiainen for their technical contributions and Justus Brown for helping us understand the requirements and business models in the domain of NFC transport ticketing. Also, the first author's work was done while being employed by Nokia Research Center, Helsinki.

12. REFERENCES

[1] R. Anderson, M. Bond, O. Choudary, S. J. Murdoch, and F. Stajano. Financial cryptography kill financial innovation? - the curious case of emv. In G. Danezis, editor, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011.

[2] ARM. Technical reference manual: Arm 1176jzf-s (trustzone-enabled processor). http://www.arm.com/pdfs/DDI0301D_arm1176jzfs_r0p2_trm.pdf.

[3] G. de Koning Gans. Analysis of the mifare classic used in the ov-chipkaart project. Master's thesis, Radboud University Nijmegen., June 2008. http://www.sos.cs.ru.nl/applications/rfid/2008-koning-thesis.pdf.

[4] G. de Koning Gans, J.-H. Hoepman, and F. Garcia. A practical attack on the mifare classic. In G. Grimaud and F.-X. Standaert, editors, *Smart Card Research and Advanced Applications*, volume 5189 of *Lecture Notes in Computer Science*, pages 267–282. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-85893-5_20.

[5] G. de Koning Gans, J.-H. Hoepman, and F. Garcia. A practical attack on the mifare classic. In G. Grimaud and F.-X. Standaert, editors, *Smart Card Research and Advanced Applications*, volume 5189 of *Lecture Notes in Computer Science*, pages 267–282. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-85893-5_20.

[6] J.-E. Ekberg and M. Kylanpaa. Mobile trusted module. Technical Report NRC-TR-2007-015, Nokia Research Center, November 2007. http://research.nokia.com/files/NRCTR2007015.pdf.

[7] EMV. *Contactless Specifications for Payment System.* Version 2.1, EMVCo, 2011.

[8] F. Garcia, G. de Koning Gans, R. Muijrers, P. van Rossum, R. Verdult, R. Schreur, and B. Jacobs. Dismantling mifare classic. In S. Jajodia and J. Lopez, editors, *Computer Security - ESORICS 2008*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-88313-5_7.

[9] F. Garcia, P. van Rossum, R. Verdult, and R. Wichers Schreur. Wirelessly pickpocketing a mifare classic card. *Security and Privacy, IEEE Symposium on*, 0:3–15, 2009.

[10] S. Ghiron, S. Sposato, C. Medaglia, and A. Moroni. Nfc ticketing: A prototype and usability test of an nfc-based virtual ticketing application. In *Near Field Communication, 2009. NFC '09. First International Workshop on*, pages 45 –50, feb. 2009.

[11] Global platform. Globalplatform card specification v2.2.1, 2011. http://www.globalplatform.org/specificationscard.asp.

[12] ISO/IEC 14443. *Identification cards – Contactless integrated circuit cards – Proximity cards.* ISO, Geneva, Switzerland, 2008.

[13] ISO/IEC 18092:2004. *Information technology – Telecommunications and information exchange between systems – Near Field Communication –*

Interface and Protocol (NFCIP-1). First edition, ISO, Geneva, Switzerland, 2004.

[14] ISO/IEC 21481:2005. *Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2)*. First edition, Geneva, 2005.

[15] ISO/IEC 7816-4:2005. *Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange*. Second edition, ISO, Geneva, Switzerland, 2005.

[16] K. Kostiainen, J.-E. Ekberg, N.Asokan, and A. Rantala. On-board credentials with open provisioning. In *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 104–115, New York, NY, USA, 2009. ACM.

[17] P. S. C. Lau. Developing a contactless bankcard fare engine for transport for london. Master's thesis, Massachusetts Institute of Technology, 2009. http://hdl.handle.net/1721.1/55337.

[18] P. Luptak. Public transport sms ticket hacking, 2009. Presented in Hacking at Random https://har2009.org/program/events/89.en.html.

[19] G. Madlmayr. Uncovered: The hidden nfc potential of the google nexus s and the nokia c7. http://www.nearfieldcommunicationsworld.com /2011/02/13/35913/uncovered-the-hidden-nfc-potential-of-the-google-nexus-s-and-the-nokia-c7/ (accessed July 2011).

[20] K. E. Mayes, K. Markantonakis, and G. Hancke. Transport ticketing security and fraud controls. *Information Security Technical Report*, 14(2):87 – 95, 2009. Smart Card Applications and Security.

[21] S. Mehta. Analysis of future ticketing scenarios for transport for london. Master's thesis, Massachusetts Institute of Technology., June 2006. http://hdl.handle.net/1721.1/34592.

[22] Smart Card Alliance. Transit and contactless financial payments: New opportunities for collaboration and convergence. A Smart Card Alliance Transportation Council White Paper, October 2006. http://www.smartcardalliance.org/resources/lib/Transit_Retail_Pmt_Report.pdf (Accessed: August 2011).

[23] J. Srage and J. Azema. M-Shield mobile security technology, 2005. TI White paper. http://focus.ti.com/pdfs/wtbu/ti_mshield_whitepaper.pdf.

[24] H. Wilcox. Mobile ticketing: Transport, sport, entertainment event 2008-2013. Technical report, Juniper Research, October 2008. http://www.juniperresearch.com/reports.php?id=155 (Accessed: July 2011).